REMARKS

In this Office Action claims 1-23 are noted as pending and rejected. In summary of this Response thereto, claims 1, 6, 11, 16, 19, 20, 21 and 23 are amended, new claim 20 is added and remarks are provided regarding the patentability of claims 1-24.

A. Specification

The Examiner has objected to the title of the invention for not being descriptive. In response thereto, Applicant has amended the title to more clearly indicate the invention to which the claims are directed.

The Examiner has also noted the use of the trademarks INTEL, ITANIUM, IBM and POWER 3. Where appropriate, Applicant has amended the specification to capitalize each trademark. It is noted that in the case where Intel and IBM or International Business Machines are used as corporate names, rather than trademarks, the original usage has been maintained.

B. Claim Objections

The Examiner has objected to claim 16 for containing informalities. More particularly, the Examiner believes the term "from" should be replace with "for". This claim has been amended according to the Examiner's suggestion.

C. Rejection of claim 20 under 35 USC Section 112, second paragraph

This claim is rejected as being indefinite for failing to particularly point out and distinctly claim the subject matter regarded by Applicant as the invention. More particularly, the Examiner states there is insufficient antecedent basis for "said status bit" in line 2. In response thereto and consistent with independent claim 19, claim 20 has been amended to recite that the retry is based on an indication included in the status register.

D. Rejection of claims 1-23 under 35 USC Section 102(e)

These claims are rejected as being anticipated by Harris (US PG Pub 2001/0056530 A1, hereinafter "Harris"). More particularly, regarding claim 1, the Examiner believes that Harris anticipates the present invention by disclosing a method of operating a data processing system having a microprocessor that executes program instructions, including determining an order of program instructions and moving a certain one of the program instructions to an advanced position in the program order. Further, the Examiner also believes that Harris checks status information associated with certain program instructions to determine whether they will execute successfully and retry the certain program instructions when they do not execute successfully.

In response thereto, it is respectfully submitted that Harris does not disclose the presently claimed invention for the following reasons. Harris at lines 1-3 of paragraph 4 states that speculative loads are often implemented as non-faulting loads. i.e. loads which always complete, even in the presence of faults. Harris goes on to state in paragraph 5, lines 7-12 that in handling a non-faulting load the software or hardware is configured such that any subsequent use of the data generates a trap and the program flow will enter into an error handling routine. Further, paragraph 6 at lines 7-15 states that when a non-faulting load returns a zero value (i.e. invalid data, paragraph 6, lines 3-4) ***"then the memory address is read again later using a normal (non-speculative) load"*** to which normal protection mechanisms apply. These normal protection mechanisms include causing an exception, i.e. generating a trap to call error handling routines.

Therefore, it is respectfully submitted that Harris discloses retrying a speculative load (non-faulting load that returns a zero value) with a normal (non-speculative) load that will utilize normal protection mechanisms (call error handling routines).

Claim 1 has been amended to more clearly recite that ***the load instruction is retried by speculatively re-executing the load*** (specification at page 18, lines 13-14, abstract lines 7-9 and 13-14). That is, after a speculative load is executed and its status is checked to determine if it has been successful then the load instruction is speculatively re-executed as recited by amended claim 1, reproduced below.

1. A method of operating a data processing system, having a microprocessor that executes program instructions, comprising the steps of:
determining an order of said program instructions;
moving a load instruction to an advanced position in said program order;
speculatively executing said load instruction;
checking status information associated with said load instruction to determine whether it will execute successfully; and
retrying said load instruction by speculatively re-executing said load instruction when said load instruction does not execute successfully;
wherein said retried load instruction does not call any recovery routines.

For these reasons, it is respectfully submitted that Harris does not anticipate amended claim 1 since all of the elements recited therein are not disclosed by this reference. Independent claims 6, 11, 16, 19, 21 and 23 have been amended in a manner similar to claim 1 and are also believed to be unanticipated by Harris for the same reasons.

Additionally, claims 1, 6, 11, 16, 19, 21 and 23 have been amended to recite that the retried load instruction does not call any recovery routines, as shown below. Harris attempts to improve microprocessor performance by delaying the testing of results from a non-faulting load (paragraph 31, lines 1-3). Paragraph 52 further teaches that upon return of a test result showing non-valid data, service routines will automatically be invoked. Thus, Harris teaches away from the presently claimed invention by delaying the checking of the load instruction validity, and continuing to call service routines when a load instruction is determined to be invalid.

1. A method of operating a data processing system, having a microprocessor that executes program instructions, comprising the steps of:
determining an order of said program instructions;
moving a load instruction to an advanced position in said program order;
speculatively executing said load instruction;
checking status information associated with said load instruction to determine whether it will execute successfully; and
retrying said load instruction by speculatively re-executing said load instruction when said load instruction does not execute successfully;
wherein ***said retried load instruction does not call any recovery routines.***

Therefore, it is believed that independent claims 1, 6, 11, 16, 19, 21 and 23 are in allowable condition as being unanticipated and non-obvious in view of 35 USC Section 102(e) and 103. Claims 2-5, 7-10, 12-15, 17-18, 20 and 22 all depend from independent claims 1, 6,

11, 16, 19 and 21, respectively and are also believed to be allowable as further defining the scope of these independent claims, which are in allowable condition, as discussed above. Thus, dependent claims 2-5, 7-10, 12-15, 17-18, 20 and 22 are also believed to be allowable under 35 USC Section 102(e) and 103 in view of Harris.

E. New claim 24

New independent claim 24 has been added which recites elements similar to claims 1, 6, 11, 16, 19, 21 and 23 as described above. In addition, claim 24 (see below) recites a load address table that stores an address of a load instruction to be speculatively executed wherein retrying speculative execution of the load instruction is based on the presence of the load address in the table (specification page 14, line 6 to page 16, line 25 and Figure 6).

24. A data processing system, comprising
a memory;
a microprocessor that executes program instructions including a load instruction which retrieves information from said memory;
a compiler that determines an order of said program instructions, and is capable of advancing said load instruction to an advanced position in said program order;
a load address table for storing an address of said load instruction to be speculatively executed by said microprocessor; and
an execution unit in said microprocessor that speculatively executes said load instruction, and determines if said load instruction executed successfully by executing an instruction to check whether said address of said load instruction remains stored in said load address table;
wherein ***speculative execution of said load instruction is retried based on the presence of said address of the speculatively executed load instruction in said load address table***; and
wherein said retried speculative load instruction does not call any recovery routines.

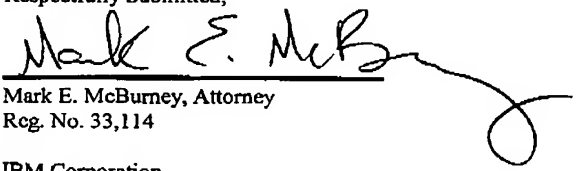
It can be seen that new independent claim 24 recites the elements of claims 1, 6, 11, 16, 19, 21 and 23, and further includes additional recited subject matter to which Applicant is entitled. Therefore, Applicant believes that claim 24 is allowable over 35 USC Section 102(e) and 103 in view of the disclosure and teaching of Harris.

F. Conclusion

For the reasons set forth above, Applicant respectfully submits that claims 1-24 are in condition for allowance.

If there are any additional fees required in connection with this Response, please charge same to Deposit Account No. 09-0447.

Respectfully Submitted,



Mark E. McBurney, Attorney
Reg. No. 33,114

IBM Corporation
11400 Burnet Road
Intellectual Property Law
Austin, Texas 78758

Phone: (512) 823-1003
FAX: (512) 823-1036
EMAIL: markmcb@us.ibm.com